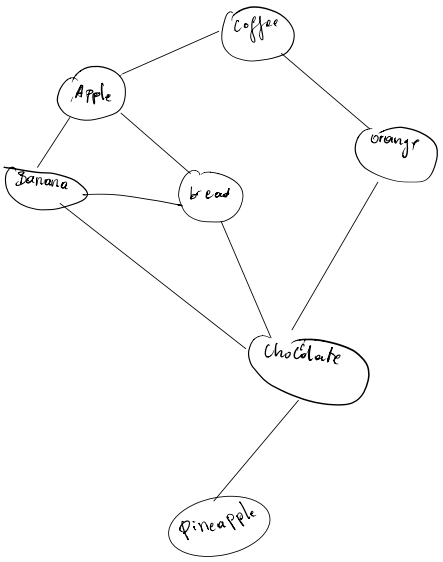
Problem Domain:

Write a breath-first traversal method that accept a starting node.

Illustration

output:



input: Apple, coffee, bread, banana, orange, chocolate, pineapple.

Algorithm:

- push the starting node in a list
- enqueu that starting node in a queue.
- run a while loop when the queue is not empty.
- dequeue the presedent enqueue node.
- Pushe all the neighborsVertex in an othe List.
- and so on.
- return the List of node.

Pseudo-code

```
Arr => [Vertex]
Queue => enqueue(Vertex)
while queue not empty:
    Vertex => dequeue
    Neighbor = [edge.vertex for edge in get_neighbors(Vertex)]
    for Ver in neigh
        if Ver not in Arr
            Arr => [Ver]
            queue => enqueue(Ver)
    Return Arr.
```